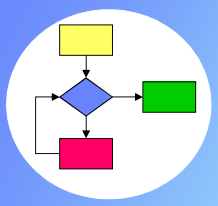
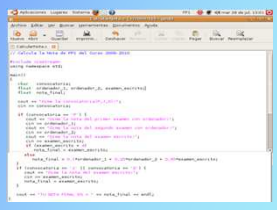


Programación 1

Tema 1

Introducción



Departamento de Ciencia de la Computación e Inteligencia Artificial

Objetivos

1. Diferenciar entre compiladores e intérpretes
2. Saber lo que es un programa y entender las fases que implica su desarrollo
3. Conocer algunos consejos para aprender a programar
4. Conocer por qué usamos lenguaje C como primer lenguaje de programación

Índice

Representación de la información

Los computadores representan la información usando dos dígitos: CODIFICACIÓN BINARIA (base 2)

BIT: (*BInary digIT*: 0 o 1) unidad de información mínima representable en un ordenador.

BYTE: 8 bits.

PALABRA: unidad mínima de tratamiento. Depende de la máquina: 1, 2, 3, 4 u 8 bytes (8, 16, 24, 32 o 64 bits)

Representación de la informacion

Un carácter se representa empleando un byte. El conjunto de caracteres codificable en un ordenador se denomina **juego de caracteres**, y está compuesto por:

- letras o caracteres alfabéticos
- dígitos o caracteres numéricos
- caracteres especiales y de puntuación
- caracteres de control (salto de línea, etc.)

JUEGO DE CARACTERES **ASCII**

(*American Standard Code for Information Interchange*)

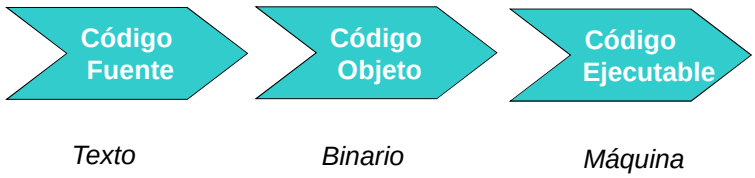
01000001-> ‘A’

Índice

Compiladores e Intérpretes

COMPILADOR

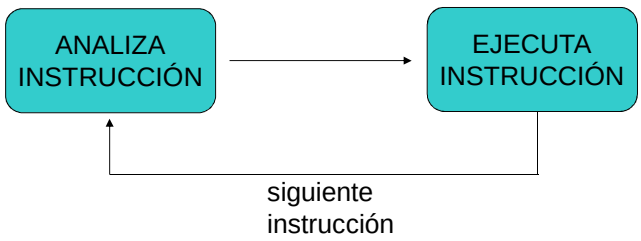
El compilador **analiza** nuestro programa comprobando su sintaxis e indicando los errores de escritura, y **genera** el programa en lenguaje máquina. Puede que necesite un **enlazado** (linkado), en donde se le unen una serie de módulos de librería.



Compiladores e Intérpretes

INTÉRPRETE

El intérprete analiza y ejecuta un programa **sentencia a sentencia.**



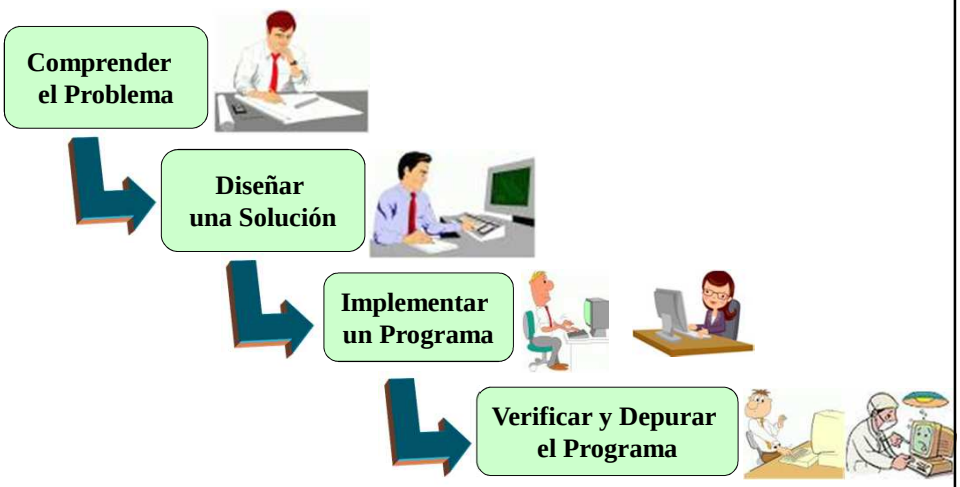
Índice

Concepto de Programa Informático

Conjunto de instrucciones ordenadas escritas en un lenguaje de programación para que un ordenador lleve a cabo una determinada tarea

Índice

¿Cómo desarrollar un programa?

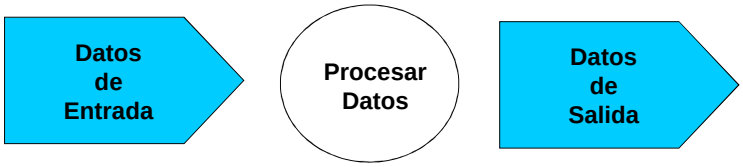


Comprender el problema

Analizar el problema :

- Responder a la pregunta…

  ¿**QUÉ** es lo que hay que resolver?



Problema: Cálculo de una nota

Problema:

Se desea calcular la nota de una asignatura de un alumno teniendo en cuenta la nota de los exámenes realizados en una determinada convocatoria.

En Enero, la nota final se obtiene con el 15% de la nota de un primer examen con ordenador, el 35% de la nota de un segundo examen con ordenador y el 50% restante con la nota de un examen escrito, excepto si la nota del examen escrito o la del segundo examen con ordenador es menor que 4, en cuyo caso la nota final será la mínima de las dos.

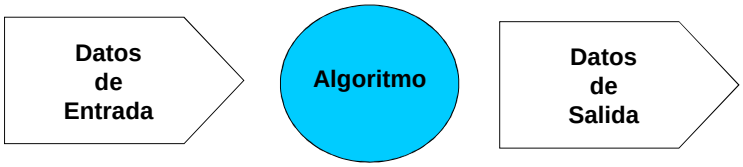
En Julio, la nota final se obtendrá como 50% nota del examen escrito, 50% nota del examen con ordenador, excepto si cualquiera de estas dos notas es menor que 4, en cuyo caso la nota final será la mínima de las dos.

Diseñar una solución

Proponer los pasos a seguir (**algoritmo**) para solucionar el problema

- Responder a la pregunta…

  ¿**CÓMO** se va a resolver?



Solución algorítmica

Algoritmo:
- Dime de qué convocatoria quieres saber tu nota final
- **Si** es la convocatoria de Enero
  Entonces
  - Dime la nota de los 2 exámenes con ordenador y del examen escrito
  - **Si** la nota del examen escrito < 4 **o** nota examen ordenador 2 < 4
    Entonces
    TU NOTA FINAL ES = mínimo (nota ex. escrito, nota ex. ord. 2)
    Sino
    TU NOTA FINAL ES =0,15*Ordenador_1+0,35*Ordenador_2+0,5*Escrito
- **Si** es la convocatoria de Julio
  Entonces
  - Dime las notas del examen escrito y del examen con ordenador
  - **Si** la nota del examen escrito < 4 **o** nota examen ordenador < 4
    Entonces
    TU NOTA FINAL ES = mínimo (nota ex. escrito, nota ex. ord.)
    Sino
    TU NOTA FINAL ES = 0,5*Ordenador+0,5*Escrito

Implementar un Programa

Codificar en un lenguaje de programación los pasos a seguir para resolver el problema :

1. Conocer la **sintaxis** de un lenguaje de programación
2. **Escribir** el programa con un editor de textos
3. **Compilar** y **corregir** errores sintácticos

Programa para calcular la nota

función que calcula el número más pequeño entre 2 números

```
#include <iostream>
using namespace std;

main()
{
   char   convocatoria;
   float  ordenador_1, ordenador_2, examen_escrito, nota_final;

   cout << "Dime la convocatoria(E,J):";
   cin >> convocatoria;

   if (convocatoria == 'E') {
      cout << "Dime la nota del primer examen con ordenador:";
      cin >> ordenador_1;
      cout << "Dime la nota del segundo examen con ordenador:";
      cin >> ordenador_2;
      cout << "Dime la nota del examen escrito:";
      cin >> examen_escrito;
      if (examen_escrito < 4 || ordenador_2 < 4)
         nota_final = minimo(examen_escrito, ordenador_2);
      else
         nota_final = 0.15*ordenador_1 + 0.35*ordenador_2 + 0.5*examen_escrito;
   }
   if (convocatoria == 'J' ) {
      cout << "Dime la nota del examen escrito:";
      cin >> examen_escrito;
      cout << "Dime la nota del examen con ordenador:";
      cin >> ordenador_2;
      if (examen_escrito < 4 || ordenador_2 < 4)
         nota_final = minimo(examen_escrito, ordenador_2);
      else
         nota_final = 0.5*ordenador_2 + 0.5*examen_escrito;
   }
   cout << "TU NOTA FINAL ES = " << nota_final << endl;
}
```

Verificar y Depurar el programa

Ejecutar el programa y corregir errores :

1. Verificar el programa (**Pruebas**)
   - Ejecutar el programa y detectar errores
2. Depurar el programa (**Depuración**)
   - Corregir errores de ejecución del programa

Datos de entrada				Datos de salida	Ok
Convocatoria	Ord 1	Ord 2	Escrito		
E	5	3	5	3	✓
E	3	6	3	3	✓
E	6	5	9	5,8	✓
P					✗
J		4	4	4	✓
J		10	5	7,5	✓
J		66	6	36	✗

Índice

Consejos para aprender a programar

- Estudiar 
- Practicar 
- Aprender de los errores 
- Buscar antes de preguntar  
- Intercambiar conocimientos y experiencias 

Se trata de un desafío, la mejor virtud de un programador es el tesón

Índice

¿Por qué utilizamos el lenguaje C?

- Es un lenguaje de propósito general
- Muy utilizado en el mundo laboral
- Facilita la programación estructurada y modular
- Utilizaremos el **lenguaje C** con algunos elementos de **C++** que facilitan el aprendizaje de iniciación a la programación

C++ bajo el paradigma **IMPERATIVO**…

… y NO bajo su paradigma natural Orientado a Objetos, que lo aprenderás en otras asignaturas de la titulación: Programación 2, Programación 3, etc.

Creación de un programa ejecutable

Necesario:

- Editor de textos -> para escribir las instrucciones. Ejemplo: Bloc de notas, kate, gedit
- Compilador y enlazador -> generar el código ejecutable. Ejemplo: g++

El proceso de compilación (código fuente-> código ejecutable) se puede realizar mediante:

- Programas basado en línea de comandos. Ejemplo: g++
- IDE (Entorno integrado de desarrollo). Ejemplo: Dev-C++, Eclipse, NetBeans. Los IDE incluyen el editor, el compilador, el enlazador y un depurador además de otros elementos.

Índice

Importancia del Análisis y Diseño

- Es fundamental comprender bien el problema antes de pensar en la solución

- Antes de ponerse a escribir el programa (implementación) es necesario tener claro cómo resolverlo.

Índice

1. Representación de la información
2. Compiladores versus intérpretes
3. ¿Qué es un programa?
4. ¿Cómo desarrollar un programa?
5. ¿Cómo aprender a programar?
6. El lenguaje C
7. Conclusiones
8. **Fuentes de información**

Bibliografía Recomendada

Fundamentos de Programación
Jesús Carretero, Félix García, y otros
Thomson-Paraninfo (2007) ISBN: 978-84-9732-550-9

- Capítulo 1 (Apartados 1.5; 1.7)
- Capítulo 3 (Apartados 3.1; 3.2)

Problemas Resueltos de Programación en Lenguaje C
Félix García, Alejandro Calderón, y otros
Thomson (2002) ISBN: 84-9732-102-2

- Capítulo 1 (Apartados 1.5; 1.7)

Resolución de Problemas con C++
Walter Savitch
Pearson Addison Wesley 2007. ISBN: 978-970-26-0806-6

- Capítulo 1